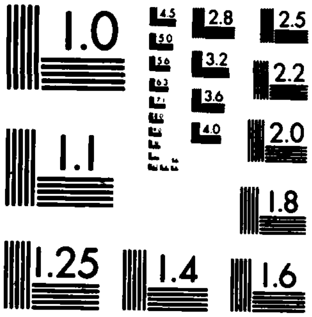
AD-A128 555 VESSEL NAVIGATION SYSTEM SIMULATION VOLUME III USER'S MANUAL(U) DELTA RESEARCH INC HUNTSVILLE AL L B WEINER ET AL. JAN 82 DR-003-020 USCG-D-03-82 1/1

UNCLASSIFIED DTCG23-80-C-20036 F/G 9/2 NL

END

DATE
FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Report No. CG-D-03-82

VESSEL NAVIGATION SYSTEM SIMULATION

VOLUME III: USERS' MANUAL

DELTA RESEARCH, INC.
2109 WEST CLINTON AVENUE
SUITE 414
HUNTSVILLE, ALABAMA 35805

FINAL REPORT

JANUARY 1982

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
United States Coast Guard
Office of Research and Development
Washington, D.C. 20590

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

The contents of this report do not necessarily reflect the official view or policy of the Coast Guard; and they do not constitute a standard, specification, or regulation.

This report, or portions thereof may not be used for advertising or sales promotion purposes. Citation of trade names and manufacturers does not constitute endorsement or approval of such products.

Technical Report Documentation Page

1. Report No.
CG-D-03-82

2. Government Accession No.
AD-A128555

3. Recipient's Catalog No.

4. Title and Subtitle
Vessel Navigation System Simulation
Volume III: Users' Manual

5. Report Date
January 1982

6. Performing Organization Code

8. Performing Organization Report No.
Delta Research, Inc.
003-020

7. Author(s)
Louis B. Weiner, Arthur Moon Jr.

9. Performing Organization Name and Address
Delta Research, Inc.
2109 W. Clinton Ave.
Suite 414
Huntsville, AL 35805

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
DTCG23-80-C-20036

13. Type of Report and Period Covered
Final Report
September 1980 -
December 1981

12. Sponsoring Agency Name and Address
United States Coast Guard
Office of Research and Development
Washington, D.C. 20590

14. Sponsoring Agency Code
G-DMT-1/54

15. Supplementary Notes

The technical representative for this contract was Dr. Ping Chuang.

16. Abstract

This Final Report discusses the mathematical formulation, computer program implementation, and examples of the Vessel Navigation System Simulation (VNSS). The VNSS simulates piloting a vessel in a restricted waterway, with channel banks, current, and obstacles. The techniques of modern optimal control theory is used to derive the control history and resultant vessel trackline, using USCG supplied code definition for vehicle dynamics.

17. Key Words
Navigation
Simulation
Optimal Control
Collision, Ramming, Grounding Risks

18. Distribution Statement
This document is available to the U.S. public through:

National Technical Information Service
Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
UNCLASSIFIED	UNCLASSIFIED	23	

TABLE OF CONTENTS

1. INTRODUCTION

The purpose of the Vessel Navigation System Simulation (VNSS) is to generate a history of rudder and RPM commands that will result in an optimum trajectory of vessel through the channel. The trajectory is optimum in the sense that the sum of squared distances from each bank divided by an approximation of the channel width is minimum. The distances are evaluated at each descrete time step of the update procedure for the vessel dynamic equations.

This program is versatile in that the initial conditions can easily be changed as well as the run length and the restart distance. It is less versatile in so far as changing the scene or the ship dynamic descriptors is concerned. What is required to change the channel definition is the information equivalent to subroutine DTN10 contained in Section 5.10.3 of the Programmers' Manual. This subroutine, which was supplied by the USCG, comprises 14 pages of listing and describes a portion of Berwick Bay Harbor. The vessel dynamic constants are contained in subroutine DTM10 and the only method of changing them that is provided is to recompile the program with a different Fortran coded DTM10 routine.

Provision has been made to either use the stored channel definition, or to input it from the keyboard or from an existing appropriately formatted data file.

This manual has been organized to describe the meaning of all variables that may be input (Section 2) followed by two different methods of doing so (Sections 3 and 4). The possibility of substituting Subroutines DTM10 or DTN10 (Section 5) is described next. Finally, the several error messages that can be generated are listed along with a description of likely causes and/or cures.

2. INPUT VARIABLE DEFINITION

Vessel Navigation System Simulation (VNSS) input can be provided by one of two methods. One of these is to respond to self-prompting messages which appear on the screen. This input sequence is described in more detail in Section 4.1. The other option is to run from a preloaded data file called DATA.DAT. The program to create new or modified input files, if this input mode is requested, is described in Section 4.2, together with a listing. While using either option, intelligent selection of these variables is necessary to the proper functioning of the algorithm.

Besides meaningful selection of input variables, there are certain, arbitrary, restrictions upon these inputs that must be enforced if proper algorithm performance is desired. The following is a list of input variable definitions and restrictions.

- NOBST - Total number of obstacles
  - If NOBST $\leq$ 0, then no obstacles are to be input
  - If NOBST > 0, then NOBST sets of obstacle coordinates must be input In this case, the Ith obstacle input is OBS(I,1), OBS(I,2), WEST(I).
- OBS(I,1) - Obstacle coordinate along the subscript 1 axis (ft)
- OBS(I,2) - Obstacle coordinate along the subscript 2 axis (ft)
- WEST(I) - Bank association indicator
  - If WEST(I) is true, then obstacle is associated with lower bank.
  - If WEST(I) is false, then obstacle is associated with upper bank.

The obstacle coordinates and bank association indicator can be input in any order. The obstacles will automatically modify the bank segments when they are within the visibility window. If, inadvertently, an obstacle is inserted outside the channel definition, it is disregarded by the code. Otherwise, the proper quadrilateral in which the obstacle is located is determined and the indicated bank is slid parallel to itself to intersect the obstacle. Bank segments are then adjusted so as to preclude 35° bank segment angular changes. If sufficient bank segment break points are not input, then the code, in enforcing this condition, may collapse the channel.

- NXY - Number of pairs of bank break point coordinates
  - If NXY $\leq$ 0, then bank segments in DTN10 are used
- CURRNT - Current indicator
  - If CURRNT is true, then current is on.
  - If CURRNT is false, then current is zero.
- XX(I,1) - Lower bank break point coordinate along subscript 1 axis (ft)
- XX(I,2) - Upper bank break point coordinate along subscript 1 axis (ft)
- YY(I,1) - Lower bank break point coordinate along subscript 2 axis (ft)
- YY(I,2) - Upper bank break point coordinate along subscript 2 axis (ft)

The bank segment subscript I must be sequentially increasing with the direction of travel.

Furthermore, the following conditions should be met:

$$XX(I+1,J) - XX(I,J) \geq 3V \qquad 2 \leq I \leq NXY,\ J=1,2$$

$$YY(I,2) > YY(I,1) \qquad 2 \leq I \leq NXY$$

where V = boat speed in feet per second.

There is also a restriction that cannot be phrased in a rigid mathematical framework. Adjacent bank segments should deviate in angle by less than 35° and pairs of channel break points should be placed roughly across the channel from each other. If violated, it does not necessarily preclude proper execution of the code or arrival at an appropriate solution. **Restriction on channel width (Minima channel width) should be discussed, i.e., channel width of 50 feet and 45° turn will result in a failure.**

Another further restriction that cannot be made rigid is that regarding channel width. If a channel width is too narrow in a turn, the inertia of the vessel may cause a collision with one or more of the bank segments. In addition, because the ship is modeled as a point mass with zero width, the banks should be input narrower than the physical channel constraints.

- VC(J) - Current magnitude (f/s)
  - J=1,8*(NXY-1)
- PSIC(J) - Current direction measured from subscript 1 axis to subscript 2 axis (degrees)
  - J=1,8*(NXY-1)

The initial angle at which the boat will travel is automatically set parallel to the first segment of the lower bank. The initial derivative of this angle is set to zero. The initial sway velocity is set to zero.

- X(1,1) - Boat initial coordinate along subscript 1 axis (ft)
- X(2,1) - Boat initial coordinate along subscript 2 axis (ft)
- U(1) - Boat initial longitudinal velocity it is with respect to inertial reference (f/s)
- PROPS - Initial propeller rates (RPM)

The initial position X(1,1),X(2,1) must be such that it is contained in the quadrilateral formed by the first upper and lower bank segments. The initial velocity, representing the resultant of the velocity of the boat and the velocity of the current, must be such that a nonnegative velocity with respect to current results.

- XWIND - Visibility or window length (ft)
- XRST - Distance between successive initial conditions if any (ft)
- XRUN - Distance of total run (ft)
- NP - Trajectory print interval
- NS - Stopping Criterion

This last variable, NS, determines the number of iterations to be performed after the trajectory lies completely within the banks. If the algorithm does not encounter three consecutive maximum number of halvings, the algorithm will continue until twice NS iterations are executed. However, if neither occur, then 100 iterations will be performed.

3. USER INTERACTIVE INPUT OPTION

On entering the program, the user may choose to input the various scene definition and program control variables described in Section 4 by means ot the program's Interactive Input Procedure (IIP). The user chooses this by answering FALSE (or F or <CR>) to the question "DØ YØU WANT TØ USE YØUR DATA.DAT FILE (T= YES, F = NØ)?"

"Interactive input" means the computer prompts the user for information and acts on his responses. Sometimes the computer prompt is a question or a request for a specific piece of information (such as a filename). In other cases, the computer presents the user with a list of possible actions *(called a menu), from* which the user chooses the course of action he wishes to take. The user continues inputting, modifying, and examining data until he is satisfied that it is all correc He may then want to store it in a file for future reference or use and, at this time, he may initiate program execution.

In the (IIP) for this program, a *set of five commands is provided to the user,* any prompt ending in a double carat ('>>'). These commands are listed in Table 3-1; they are also described to the user when entering the IIP. The commands make it possible to examine data, skip over irrelevant data, and retain default values where desired.

IIP CØMMANDS

'X'	EXIT THIS MENU (GØ TØ NEXT HIGHER MENU)
'W'	WRITE ØUT DEFAULTS (PRESENT VALUES)
<CR> OR ' '	KEEP DEFAULTS FØR THIS PØINT, GØ TØ NEXT
'N'	KEEP DEFAULTS FØR THIS PØINT AND REMAINING PØINTS IN THIS GRØUP (E.G., SKIP TØ NEXT BANK ØR CURRENT STATIØN)
'H'	WRITE AN EXPLANATION ØF THESE CØMMANDS AND RELATED MATERI ØN THE SCREEN

Table 3-1

Detailed explanation of expected input precedes many of the prompts, although the user may cancel this once he has become familiar with the IIP by responding FALSE to the LØNGPRØMPTS> prompt. In general, user input will consist of integer, real, and logical constants, with no internal blanks, separated by commas. The required data types are usually exemplified by the variable names (which follow the FORTRAN default name typing conventions) and present default values.

To avoid confusion, texts mention logical variables explicitly since logical variables have no default name types. If the user should enter a value of the wrong type, Fortran will report the error, after which an explanatory note will appear which includes a list of the expected data types for the required variables. This list is called a "template." The IIP uses five such templates (See Table 3-2), and by noting the template number of the bad read, the user may enter the correct values when he returns to the prompt (immediately after the explanatory note). While this takes some of the pressure off the user (he won't lose all of his work because of one careless error), care must be taken not to exceed the error count, as the program will die then no matter what.

Table 3-2. IIP TEMPLATES

TEMPLATE 1	L	ØNE LØGICAL VALUE
2	I	ØNE INTEGER VALUE
3	R,R	TWØ REAL VALUES
4	I,I,R,R	TWØ INTEGERS, TWØ REALS
5	R,R,L	TWØ REALS, ØNE LØGICAL

The prompts given by the IIP are virtually self-explanatory. The super structure of the IIP is embodied in subroutine UINPUT (see Section 5) to which the interested user or programmer is referred.

4. AUXILLIARY INPUT ROUTINE

In the likely event that VNSS is to be run a number of times with an almost identical input set, the auxilliary program CGIN can be most useful.

This program is composed of three sequential portions. The first portion reads the formatted input data sequence from a file named DATA.DAT and prints these values on the screen.

The second portion prompts the user to change the values or the number of entries in this file. The third portion rewrites the thus modified input data sequence to a new version of the DATA.DAT file. This new version can be selected to be the input data for a subsequent run if the VNSS program is desired.

This program also creates a semi-permanent file of the input data used which is helpful if a previous run is ever to be recreated.

5. SUBROUTINE INTERCHANGEABILITY

A possibility exists that the USCG has now, or will have in the future, various vessel and scene descriptions in the format of the supplied subroutines DTL10, DTM10, and DTN10. The VNSS program has been written in such a fashion that any or all of these subroutines may be interchanged with correspondingly named ones.

An obvious restriction is that all named common blocks must be dimensioned in exactly the manner as the supplied subroutines. Other less obvious restrictions also exist. Among these is the appropriate setting of variable NPTL in MAIN. Currently, it is set to the numeric 30 which is appropriate for DTN10 corresponding to Berwick Bay. The required change would be to set it equal to the variable NCCT which appears in common block, B9.

The VNSS program requires that the vessel travels in the general direction of increasing X and increasing bank breakpoint subscripts. In the event that a desired run for an interchanged DTN10 does not have this property, then a coding change must be made.

Also required is that the upper bank be referenced in polar coordinates from the lower bank. That is, so that the upper bank breakpoints will be designated by subscript 2. When such is not the case, it must be remedied by an appropriate coding change.

6. OUTPUT MESSAGES

6.1 Messages Appearing on S⁻reen

- *BOAT CROSSES BANK*

  This message indicates that one or more of the trajectory points lies outside the channel. This message is typical during the early iterations of the converging process. Once this message fails to appear before the iteration counter, then it should never reappear again for the current trackline.

- *NIT=* number

  This is the iteration counter.

- *SEARCH TYPE*=number

  If 0--indicates a doubling of rudder command increment takes place

  If 1--indicates a halving of rudder command increment takes place

  If 2--indicates first search point where maximum rudder command increment is 3 miliradians.

- *UNUSUAL CHANNEL CONDITION - BEWARE!*

  This message indicates that the bank subscript numbers used in the objective function calculation differ by five or more. This may be normal for a particular channel; it may indicate that a temporary error has been made, or it may indicate that an error from which there is no recovery has been made. The message is preceded by the subscripts of the lower bank followed by the upper bank (subscript 2 refers to the first segment, 3 to the second, etc.). If, in relation to the channel under investigation, the vessel cannot be in a position where it is 'keying' off the output subscripts, then the program should be aborted and an inspection of the input banks breakpoints made. This error is usually the result of consecutive X - coordinates spaced too closely together (See Section 2).

6.2 Messages Appearing on Printed Output

- *BOAT CROSSES BANK*

  See Section 6.1

- *NIT =*

  See Section 6.1

- *UNUSUAL CHANNEL CONDITION--BEWARE!*

  See Section 6.1

- *ERROR - CHANNEL DEFINITION EXCEEDED*

  This message is generated if the run length causes the vessel trajectory to extend beyond the end of the channel definition. If it occurs during the early iterations of the converging process, but not the later ones, then it can be disregarded. When it occurs, at the later iterations, then the run length input, XRUN, was unnecessarily large. Due to a late retrofit, this condition causes no problem other than extended execution time.

7. COMMAND FILES

To facilitate user manipulation (when required) of the simulation, certain command and related files have been provided. Their names are:

- REV3.TSK = the simulation itself
- REV3.CMD = compiles all routines for, and task builds, the simulation
- REV3T.CMD = task build directives file
- REV3Ø.ØDL = overlay description file
- REV3P.CMD = spool a complete listing of the simulation and its ancillary files.

Therefore, the following commands and command sequences can be used to accomplish certain user goals:

- To initiate program execution:

  ```
  >RUN REV3
  ```

- To task build:

  ```
  >TKB @REV3T
  ```

- To recompile a single routine and run:

  ```
  >FOR name = name
  >TKB @REV3T
  >RUN REV3
  ```

- To recompile all routines and run:

  ```
  >@REV3
  >RUN REV3
  ```

- To spool a complete listing (listing contains UIC directory, individual routine listings, command file listings, overlay description file listing, task build map, and run results):

  ```
  >PIP
  PIP>@REV3P
  PIP>^Z
  ```

- To spool a short listing (similar to above, but without individual routine listings):

  ```
  >PIP
  PIP>@REV3SP
  PIP>^Z
  ```

Listings of the important files mentioned in this section follow.

7.1 REV3.CMD

```
;
;THIS COMMAND FILE COMPILES ALL SUBROUTINES FOR THE NSS USING THE
;  FOR COMPILER, AND THEN TASK BUILDS THE NSS.
;
;  NOTE THAT [333,3] CONTAINS NSS, REV. 3.1 (PRELIMINARY VERSION),
;  WHEREAS [333,2] CONTAINS NSS, REV. 3.0 (EARLY REVISION 3; IE,
;  AN INTERMEDIATE RATHER THAN A FINAL VERSION).
;
TIME
FOR MAIN,MAIN/-SP=MAIN
FOR BANKS,BANKS/-SP=BANKS
FOR CLS,CLS/-SP=CLS
FOR CURT,CURT/-SP=CURT
FOR DEGRE,DEGRE/-SP=DEGRE
FOR DRU10,DRU10/-SP=DRU10
FOR DRV10,DRV10/-SP=DRV10
FOR DTL10,DTL10/-SP=DTL10
FOR DTM10,DTM10/-SP=DTM10
FOR DTN10,DTN10/-SP=DTN10
FOR DYNA2,DYNA2/-SP=DYNA2
FOR DYN2,DYN2/-SP=DYN2
FOR FW1DWN,FW1DWN/-SP=FW1DWN
FOR HDYN,HDYN/-SP=HDYN
FOR HELP,HELP/-SP=HELP
FOR INIT,INIT/-SP=INIT
FOR P,P/-SP=P
FOR PIER,PIER/-SP=PIER
FOR PROMPT,PROMPT/-SP=PROMPT
FOR PROP,PROP/-SP=PROP
FOR RES10,RES10/-SP=RES10
FOR SUB11,SUB11/-SP=SUB11
FOR SUB12,SUB12/-SP=SUB12
FOR TRACK,TRACK/-SP=TRACK
FOR UCNTRL,UCNTRL/-SP=UCNTRL
FOR UINPUT,UINPUT/-SP=UINPUT
FOR UREAD,UREAD/-SP=UREAD
FOR UREADI,UREADI/-SP=UREADI
FOR UWRITE,UWRITE/-SP=UWRITE
FOR U1,U1/-SP=U1
FOR U1DFLT,U1DFLT/-SP=U1DFLT
FOR U2,U2/-SP=U2
FOR U2DFLT,U2DFLT/-SP=U2DFLT
FOR U3,U3/-SP=U3
FOR U3DFLT,U3DFLT/-SP=U3DFLT
FOR U4,U4/-SP=U4
FOR U4DFLT,U4DFLT/-SP=U4DFLT
FOR WIDTH,WIDTH/-SP=WIDTH
FOR X,X/-SP=X
;  TKB @REV3T
TIME
```

7.2 REV3T.CMD

```
REV3/FP,REV3/-SP=REV3D/MP
ASG=SY:1:2:5:6,TI:3:4
STACK=512
//
```

7.3 REV3∅.∅DL

```
            .ROOT  MAIN-X-UCNTRL-DEGRE-PTER-*(F1,F2)
F1:         .FCTR  UINPUT-*(F11,F12)
F11:        .FCTR  INIT-DTL10-DTM10-DTN10-SUB11
F12:        .FCTR  F12A-F12B-F12C
F12A:       .FCTR  U1-U2-U3-U4
F12B:       .FCTR  U1DFLT-U2DFLT-U3DFLT-U4DFLT
F12C:       .FCTR  PROMPT-UREADI-HELP-UREAD-UWRITE
F2:         .FCTR  TRACK-P-F2A-*(F21,F22)
F2A:        .FCTR  HDYN-CURT-DRU10-PROP-DRV10-FW10MN-RES10
F21:        .FCTR  BANKS-DYN2
F22:        .FCTR  DYNA2
            .END
```

7.4 REV3P.CMD

```
;
;THIS FILE SIMPLIFIES PRINTING OUT FILES RELATED TO THE HSS (REV 3.1).
;  TO USE IT, TYPE "PIP", AND RESPOND TO THE PROMPT WITH "@REV3P".
;  AFTER THE PRINT HAS BEEN QUEUED, PIP WILL PROMPT AGAIN; RESPOND
;  WITH A "^Z" TO EXIT PIP.
;
JUNK.DAT;*/DE
JUNK.DAT=*.*;*/LI
JUNK.DAT,REV3P.CMD,REV3.CMD/SP
MAIN.FTN,BANKS.FTN,CLS.FTN,CURT.FTN,DEGRE.FTN,DRU10.FTN/SP
DRV10.FTN,DTL10.FTN,DTM10.FTN,DTU10.FTN,DYNA2.FTN/SP
DYN2.FTN,[illegible].FTN,HDY1.FTN,HELP.FTN,INIT.FTN,P.FTN/SP
PTER.FTN,PROMPT.FTN,PROP.FTN,RES10.FTN,SUB11.FTN,SUB12.FTN/SP
TRACK.FTN,UCNTRL.FTN,UINPUT.FTN,UREAD.FTN,UREADI.FTN,UWRITE.FTN/SP
U1.FTN,U1DFLT.FTN,U2.FTN,U2DFLT.FTN,U3.FTN,U3DFLT.FTN/SP
U4.FTN,U4DFLT.FTN,WIDTH.FTN,X.FTN/SP
REV3T.CMD,REV30.ODL/SP
```